

جامعة طرابلس
كلية تقنية المعلومات
قسم نظم المعلومات

المقرر الدراسي ITGS124

تحليل وتصميم نظم

Systems Analysis and Design

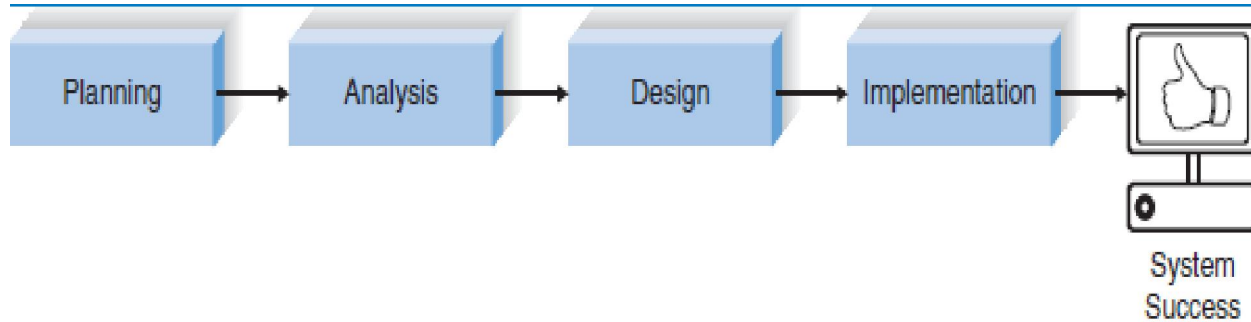
المحاضرة الأولى

أ.فاطمة بشير القاضي
algadyfatma@gmail.com

دورة حياة تطوير البرمجيات (النظم)

Software Development Life Cycle (SDLC)

تتكون دورة حياة تطوير البرمجيات بشكل عام من عدة مراحل متتابعة:



1. التخطيط Planning
2. التحليل Analysis
3. التصميم Design
4. التنفيذ Implementation

بالإضافة إلى المراحل الأساسية (التخطيط - التحليل - التصميم - التنفيذ) وبعد الانتهاء من مرحلة التنفيذ، تأتي المراحل التالية:

- الاختبار Testing
- الصيانة Maintenance

1. مرحلة التخطيط Planning

تتكون مرحلة التخطيط من أربع أنشطة رئيسية هي:

(1) طلب المستخدم User Request

تكون عملية تطوير النظام المعلوماتي نتيجة حاجة المستخدم او نتيجة مشكلة تؤثر على عمل مؤسسة ما .تستخدم المؤسسة نموذج خاص لطلب المستخدم لغرض تحديد مشاكل النظام الحالي و الحدود والأهداف المرجوة من النظام المقترح. يطلب المستخدم فى هذا النموذج احد الخيارات وهى إنشاء نظام جديد أو تحسين وتطوير النظام الحالي.

(2) دراسة الجدوى Feasibility Study

يتم عادة إعداد دراسة جدوى للمشاريع الكبيرة والمتوسطة. تتكون الجدوى من:

الجدوى الاقتصادية Economic feasibility: ويقصد بها تحديد التكلفة (مثل المعدلات المادية والبرمجية - العناصر البشرية) والمزايا (تحسين الخدمات مثل الدقة وسرعة انجاز العمل) يتم إعداد جدول لمقارنة التكلفة التقديرية لبناء المشروع مع الفوائد المتوقعة.

تابع مرحلة التخطيط

الجدوى الفنية (التقنية) Technical feasibility: تتعلق الجدوى الفنية بإمكانية

توفير المعدات المادية والبرمجية للمشروع. كذلك توفير خبرات فنية للمشروع.

الجدوى التشغيلية Operational feasibility: هي دراسة تهدف لتحديد الاتي:

- هل سيكون النظام سهل الاستخدام؟
- هل سيعمل النظام بكفاءة؟
- هل سيدعم المدراء والمستخدمون النظام الجديد؟
- هل يتطلب تدريب المستخدمين على إتقان استخدام النظام الجديد؟

(3) وثيقة خطة المشروع Project Plan Document

تعد وثيقة خطة المشروع بعد دراسة الجدوى وهي تشمل الاتي:

- ملخص المشروع
- الجدول الزمني لبناء المشروع
- الوظائف المطلوبة
- المستلزمات المطلوبة
- الاهداف المرجوة
- التكلفة التقديرية

تابع مرحلة التخطيط

4 وثيقة مقترح المشروع Project Proposal

تعرض هذه الوثيقة على المسؤولين او الإدارة لغرض الموافقة والاعتماد. تتضمن وثيقة المشروع الاتى:

- مشكلة النظام الحالى
- شرح النظام المقترح
- الجدوى الاقتصادية والفنية والتشغيلية للنظام المقترح



2. مرحلة التحليل Analysis

يتم في هذه المرحلة تحديد (المتطلبات الوظيفية) وهى تعنى الوظائف التي سيؤديها النظام المقترح، وأيضا (المتطلبات الغير وظيفية) مثل سهولة الاستخدام والأمن. نشاطات مرحلة التحليل:

- إيجاد الحقائق وجمع المتطلبات **Requirements gathering**: الخطوة الأولى في مرحلة التحليل هي فهم كيف يعمل النظام الحالي والمشاكل التي يعاني منها ومن ثم جمع واستخلاص المتطلبات للنظام المطلوب تطويره. يستخدم محلل النظام **System Analyst** عدة طرق لجمع المتطلبات منها:
 - البحث **Research**: عمل بحث واستقصاء عن النظام باستخدام تسهيلات مثل المكتبة وشبكة الانترنت.
 - الاستبيان **Questionnaire**: أسئلة ذات علاقة بالنظام تعطى للمستخدم مع تعليمات واضحة للإجابة عليها.
 - العينات **Sampling**: جمع عينات من النماذج والتقارير الخاصة بالنظام الحالي والتي يمكن أن تساهم في فهم عمل النظام الحالي وتحديد مدخلاته ومخرجاته.

تابع مرحلة التحليل

- المقابلات Interview: إجراء مقابلات مع المستخدمين والزبائن لفهم دور كل منهم في النظام الحالي والمشاكل التي يواجههم والمتطلبات المرغوب توفرها في النظام المقترح.
- النموذج التجريبي Prototype: اعداد نظام تجريبي يتم عرضه على المستخدم لتلقى ملاحظات تساهم في فهم متطلباته.
- الملاحظة Observation: مشاهدة النظام الحالي واقعيا لفهم كيفية عمله.

• تحليل المتطلبات Requirements Analysis

بعد تجميع المتطلبات بالطرق التي تم عرضها مثل المقابلة والاستبيان يجب ان تنظم للوصول الى فهم أفضل للنظام القائم (الحالي) والنظام المقترح وتصبح اساس لمرحلة التصميم.

في المنهجية الهيكلية، يستخدم المحلل مجموعة أدوات لتحليل المتطلبات مثل:

✓ مخطط انسياب البيانات Data Flow Diagram (DFD)

✓ الانجليزية المركبة او الهيكلية Structured English

في نهاية مرحلة التحليل يقوم المحلل بإعداد وثيقة تسمى وثيقة توصيف المتطلبات.

3. مرحلة التصميم Design

يتم في مرحلة التصميم تمثيل الحل (النظام المقترح) باستخدام مخططات وواجهات مستخدم وخوارزميات. تتكون مرحلة التصميم من عدة مهام تتضمن:

- تصميم واجهات الاستخدام User Interface Design: التفاعل بين النظام والمستخدم والمتمثل في تصميم شاشات الإدخال والإخراج وتصميم التقارير والنماذج الصادرة عن النظام.

- التصميم المعماري Architectural Design: يقصد به تجزئة النظام المراد تطويره إلى مكونات (أجزاء برمجية) في شكل هرمي، والهدف هو تبسيط النظام لتسهيل عملية تصميم الخوارزميات.

من أشهر أدوات التصميم المخطط الهيكلي Structure Chart وهو عبارة عن أداة تصميم رسومية توضح تركيبة النظام في شكل هرمي. يبين هذا المخطط الهيكلية الكاملة للنظام كأجزاء برمجية modules (مثل ادخال بيانات طالب، ادخال بيانات مادة، بحث عن طالب وهكذا) كذلك يبين المخطط البيانات التي تمرر بين الاجزاء البرمجية.

تابع مرحلة التصميم

- تصميم البيانات Data Design: تعتبر عملية تصميم البيانات من الأنشطة المهمة عند تطوير النظام. تبدأ عملية تصميم البيانات من مرحلة التحليل حيث يتم التعرف على البيانات التي تخص النظام ومن ثم تنظيمها وهيكلتها لتسهيل عملية تخزينها واسترجاعها ومعالجتها.

يتم تحديد الكائنات الخاصة بالنظام وخصائص كل كائن والعلاقة بين الكائنات. يستخدم مخطط كائن علاقة Entity Relationship Diagram (ERD) لتصميم البيانات حيث يستخدم هذا المخطط لوصف الكائنات وخصائصها والعلاقة بينها (يستخدم في تصميم قواعد البيانات العلائقية).

- تصميم الأجزاء البرمجية و الخوارزميات:
تصمم الأجزاء البرمجية باستخدام أدوات مثل الشفرة المزيفة Pseudo Code حيث تستخدم الشفرة المزيفة لتوضيح منطق عمل الجزء البرمجي بلغة دقيقة ومفهومة باستخدام تركيبات البرمجة الشرطية والتكرارية.

(Read, Print, if, loop, End)

4. مرحلة التنفيذ Implementation

تتم في مرحلة التنفيذ كتابة شفرة المصدر Source Code للنظام. وتسمى أيضا مرحلة البرمجة Coding. تترجم مواصفات التصميم للعمليات المختلفة للنظام الى شفرة. تستخدم لغة برمجة مثل فيجول او Java او لغة تطوير وبرمجة تطبيقات الويب مثل PHP لبرمجة النظام. وكذلك تنفيذ قاعدة بيانات النظام بأحد برامج قواعد البيانات مثل SQL Server او MySQL.

في مرحلة التنفيذ او البرمجة يجب تحقق الاتي:

- اختيار لغة برمجة للتنفيذ تناسب النظام.
- جعل المخرجات واضحة ومفهوم للمستخدم باستخدام الواجهات الرسومية.
- جعل شفرة المصدر قابلة للقراءة باستخدام أسماء ذات معنى واستخدام التعليقات.
- جعل الذاكرة المطلوبة اقل ما يمكن.

مرحلة الاختبار Testing

الهدف الرئيسي من الاختبار هو تحديد إمكانية الحصول على النتائج المطلوبة عند تشغيل النظام بدون أخطاء.

اختبار النظام الجديد يجب أن يحقق الآتي :

- اكتشاف الأخطاء التنفيذية ومعالجتها.
 - تحديد أخطاء مدخلات المستخدم.
 - تقييم سرعة أداء النظام.
 - تقييم أمانة النظام.
 - اكتشاف أي عملية أو وظيفة لم يتم تنفيذها (برمجتها).
- تستعمل عادة عينة من البيانات لاختبار النظام الجديد وتُقارن المخرجات (النتيجة) مع النتيجة المتحصل عليها من نفس العينة يدوياً. يجب أن تحتوي العينة على جميع الحالات المحتملة للتأكد من عدم وجود أخطاء. يتم اختبار كل عملية (مثل إدخال بيانات – بحث – عرض تقرير وغيرها) بشكل منفصل قبل اختبار النظام ككل.

تابع مرحلة الاختبار

هناك عدة اختبارات أهمها:

- اختبار التكامل Integration Test: يتم في هذا النوع ربط واختبار الأجزاء التي تم اختبارها بشكل منفرد كنظام متكامل للتأكد من إن المتطلبات التي عرفها المستخدم قد تم تحقيقها.
- اختبار القبول Acceptance Test: يتم تنفيذه من قبل المستخدم لإثبات إن المتطلبات التي تم تحديدها في وثيقة المتطلبات قد تم انجازها وان المستخدم موافق على النظام وان النظام جاهز للعمل.



الصيانة Maintenance

تعتبر صيانة البرمجيات في دورة حياة النظام البرمجي المرحلة الأطول على الإطلاق، إذ لا تنتهي طوال فترة عمل البرمجية؛ أي أنها دائمة حيث تعمل على جعل البرمجية مواكبة لأحدث التطورات والمعدات الجديدة بإجراء التعديلات عليها وتصحيح الأخطاء، وكما تشد مل أيضاً على إمكانية إضافة تقنيات جديدة للبرمجية.

حيث يمكن تعريفها بأنها جميع التعديلات التي تطرأ على البرمجية بعد الانتهاء منها وتسليمها للزبون لغايات تصحيح الأخطاء الموجودة أو العمل على رفع مستوى أداء البرمجية وتحسينها.

